DØ/TRIGGER/3645
Version 0.2.5
September 18, 2002

Specification of the Level 2 Central Tracking Trigger Preprocessor Crate

Drew Baden, Christopher Innes, and Dave Toback

University of Maryland

Fred Borcherding and Kin Yip

Fermi National Accelerator Laboratory

Brad Abbott, Phillip Gutierrez, Daniel Mihalcea and Miroslav Kopál

Oklahoma University

Abstract

We present a technical design report of the Level 2 Central Tracking Trigger (L2CTT) preprocessor crate. This crate will take input from the Level 1 Central Fiber Tracker (L1CFT). We include a full specification of the inputs and outputs of the crate as well as algorithms used for processing. The design is felixible enough to allow inputs from the Level 2 Silicon Tracking (L2CTT) instead of the L1CFT, with only minor modifications. This document only addresses the scenario of taking in data directly from the L1CFT.

Contents

1 Introduction

This note is the technical design report for the Level 2 (L2) Central Tracking Trigger (L2CTT) preprocessor crate. It includes a full specification of the inputs and outputs of the crate as well the algorithms used for processing. While this document is designed to be stand-alone, it only summarizes the functioning of the general L2 preprocessor as it relates to the L2CTT and leaves the details to the references [1, 2].

1.1 Overview

A schematic overview of the L1 and L2 trigger hardware is shown in Figure 1. The L2 hardware consits of a number of preprocessor crates all connected to a centralized decision making crate, the L2 global (L2G). The L2CTT is one of these preprocessor crates. The L2CTT takes input from the Level 1 Central Fiber Tracker (L1CFT). This document will focus on the L2CTT specifically, whereas the L2G and the general functioning of the L2 preprocessors is described in detail elsewhere [1]. Within this system, L2 is being designed to handle an average input rate of about 10 kHz, provide a rejection factor of approximately 10, yield a signal efficiency no lower than Run I Level 3 (L3), introduce at most 5% dead-time, and send information to the L3 trigger to assist the software filter decisions. It is important to note here that simulations have estimated that the mean processing time plus output formatting time for each preprocessor must be less than 50 μsec.

Later in the run the L2CTT will have to take input from the L2 Silicon Tracking Trigger (L2STT). So the L2CTT must operate in two different scenarios:

1. With input tracks straight from the Level 1 Central Fiber Tracker (L1CFT).

2. With input tracks from the L2STT which receives input from the L1CFT.

This document describes the L1CFT input scenario, with the L2STT input scenario described in xxxx . In the first scenario (before the L2STT is installed) the L2CTT performs the following tasks:

- Reads in the track lists from different ϕ regions of the L1 tracking trigger system.
- Converts the L1 tracking information into L2 "objects."
- Concatenates the track lists into a single list and sorts it by P_T.
- Evaluates ϕ_0 and extrapolates this value to the third layer of the calorimeter (ϕ_{em3}).
- Evaluates the CFT isolation properties of each track.
- Sends the ordered list of L2 objects to the L2G.
- Maintains L2 Objects for L3 readout on a L2 Accept (L2A) or for min-bias and monitoring triggers.

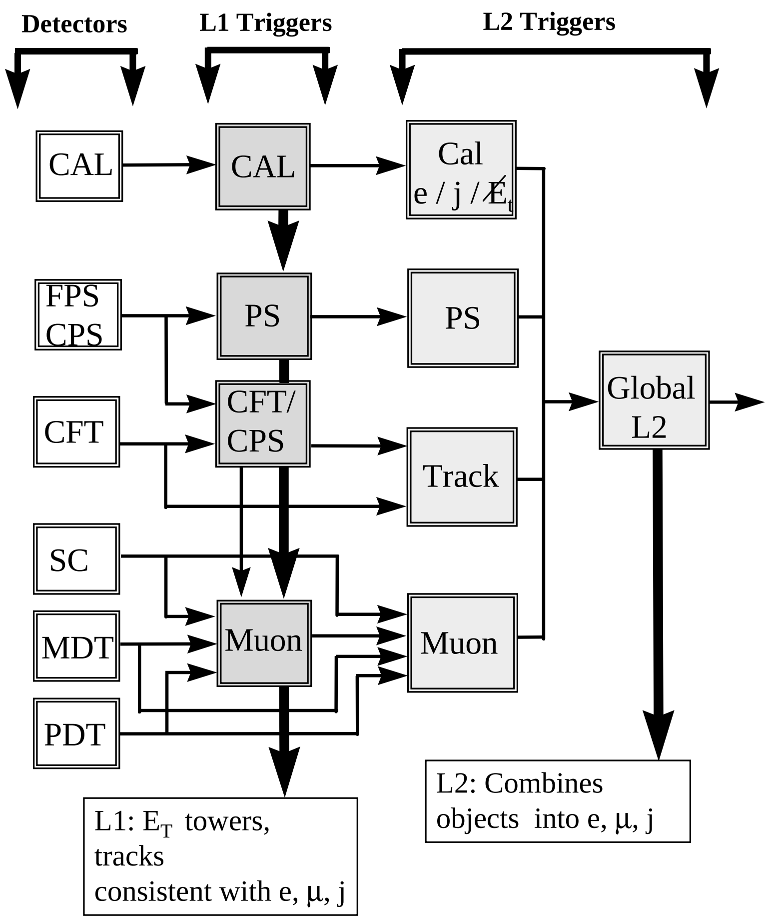

Figure 1: Overview of the L1/L2 hardware trigger system (before the L2STT comes online).

2 Functional Description of the L1 and L2 Tracking Trigger System

We begin with a more detailed description of the L1 and L2 triggers with an emphasis on the involvment of the L2CTT. In the next section we will describe the hardware implementation of the L2CTT functions described here.

2.1 The CFT and the L1 Trigger System

The CFT detector consists of 32 concentric barrel-shaped layers of scintillating fibers arranged in 16 “doublet” layers. The fibers in each layer are at the same radial distance from the z-axis and are such that half of the layers have fibers parallel to the z-axis and half are at an angle to the z-axis to provide stereo tracking reconstruction. In the ϕ coordinate, the fiber tracker system is organized into 80 sectors, each of which subtends about 4.5 degrees. The inner “A” layer and outer “H” layer are spanned by 16 and 44 fibers per sector respectively.

The L1 tracking trigger performs the following tasks:

- Reads out the tracking hits from the CFT and does the track finding.
- Reads out preshower hits from the Central Preshower (CPS) detector and does cluster finding.
- Matches CFT tracks with CPS clusters.
- Stores CFT tracks and CPS clusters for use in the event of a L1 Accept (L1A) or a L2A. In the event of a L1A, the track lists are read out to L2. In the event of a L2A, the track lists are read out to L3.

During normal operation the trigger begins by reading in the raw fiber hit information from the CFT using a set of Front End (FE) boards. A single FE board covers a 9 degree section (two sectors). All tracks with $P_T > 1.5$ GeV which cross the outer detector layer ($|\eta_{\mathrm{Detector}}| < 1.5$) are completely contained within one sector (the home sector) or are partly in one of the adjacent sectors (neighbor sectors). To form a seamless trigger, all outer layer hits from the home sector are combined with the hit information from the two neighboring sectors. Track finding uses only the axial layers and is implemented by comparing hit patterns to pre-programmed possible track roads with $P_T > 1.5$ GeV. In parallel, the CPS does its cluster finding. After track and cluster finding, there is an attempt to match each track and cluster candidates [2].

After track finding, each track candidate is classified using a 16-bit word (including P_T and ϕ at the H layer as well as the cluster matching information) and sorted into one of four P_T bins. The lowest P_T bin, Bin 3, is such that 3.0 GeV$\geq P_T^{\mathrm{Bin3}} > 1.5$ GeV. Similarly, Bins 2 through 0 are such that 5.0 GeV$\geq P_T^{\mathrm{Bin\ 2}} > 3.0$ GeV, 10.0 GeV$\geq P_T^{\mathrm{Bin\ 1}} > 5$ GeV and 10.0 GeV$< P_T^{\mathrm{Bin\ 0}}$.

After binning, the tracks are read out of each of the 40 FE boards. A maximum of 24 tracks (6 max per P_T bin) are sent from each FE board to one of eight CFT/CPS Collector boards. This means that each Collector board receives information from 10 sectors (one octant). Each Collector board then passes a maximum of 24 tracks, selected by P_T bin order, but not by P_T within the bin, to the the L2 Broadcaster (L2CFTBC) board[1]. Two Collector (CFTCPSCOL) boards connect to one Broadcaster board. This means there must be four Broadcaster boards and so each of these covers a quadrant of 20 sectors each.

A few notes about the algorithm used for selecting tracks for output to the Broadcaster boards are in order. The algorithm begins by reading all the Bin 0 tracks from all its corresponding FE boards begining with the lowest ϕ. If less than 24 tracks are read in, the algorithm reads in tracks from Bin 1 until either 24 tracks are reached or all tracks are entered into the output list. This continues for the rest of the P_T thresholds (Bins 2 and 3) until either all tracks are read in or all 24 track slots are used. It should also be noted that there is an inherent ϕ bias in this procedure as tracks with large ϕ could be lost. Similarly, it means that low P_T tracks could be lost. However, this has been studied and the current belief is that the bias is negligible [2].

When the Broadcaster card recieves its data from the Collector boards, it merges the $P_T$bins from the two Collector boards. For instance, Bin 3 from the first Collector board is merged with bin 3 from the second Collector board. Then the list that will be sent to L2 is truncated to 46 tracks although all tracks are stored to be read out to L3 in the case of a L2A. The next thing to occur is that the list is sent from the L2CFTBC to the L2CTT, where the list accompanied by the corresponding header and trailer information. This transmission is via G-links as 16-bit frames at 53 Mhz (1 Gbit/sec). As a conservative estimate, it is expected that data will be fully transferred to L2 within 2.0 μs after a L1A. This must be realized in practice as there are no output buffers on the L1CFT.

2.2 CTT: Level 2

At Level 2, there is no independent CFT trigger. For every event the tracks are read into the L2CTT, processed into final format and passed on to the L2G. There the tracks are associated with muon, electron and tau candidate information from the rest of the detector and used to make a final L2 decision. The L2CTT performs the following tasks:

- Reads in the track lists from the four L2CFTBC boards.
- Converts the L1 information to P_T.
- Uses the ϕ position at the H layer and the P_T reported by L1CFT to determine ϕ_0. The quantity ϕ_0 represents the initial ϕ of the track in the global coordinates of the detector (i.e., ϕ_0 is the ϕ angle that the track had when it left the center of the detector.
- Uses the ϕ position at the H layer and the P_T reported by L1CFT to extrapolate to the expected ϕ_{EM3}. The quantity $\phi_{EM3}$is the ϕ-angle (in the global coordinates of the detector) that the track particle will have when it reaches the third layer of the Electromagnetic Calorimeter (abbreviated EM3 but also called Showermax).
- Determines the isolation properties of each track (l2isolation). This variable will enhance the trigger efficiency in detecting tau leptons. [3]

[1]Although it is called the Level 2 Broadcaster it is part of Level 1, not Level 2.

- Concatenates the four lists of tracks into a single list which is then sorted by P_T.
- Converts the information for each track into L2 track objects.
- Sends out all tracks to L2G sorted by P_T.
- Stores information for readout on L2A or for min-bias and monitoring triggers.

To enhance performance, new events from L1CFT are allowed to flow into buffers in the crate while processing on previous events takes place. The remainder of this document describes the L2CTT.

3 CTT Preprocessor Crate Hardware

The L2CTT is implemented using the components of a standard L2 crate [1]. A schematic of the components in the crate is shown in Figure 2. For a more specific listing of which board is in which slot in the crate see Table 1. Specifically, the L2 crate is a VME64 bus crate with an additional custom bus added on the backplane below the VME bus. This is the Magic Bus (MBus) and it is an asnychronous 128 bit data/32 bit address bus. We quickly describe the cards in the L2CTT crate:

- Fiber Input Converter (FIC, see Ref. [4]): The purpose of the FIC is to convert from the G-Links connection protocol to the Cypress Hot Links connection protocol. The FIC has two parts: 1) a backplane card which has the input connectors (G-Link) and 2) a card in the front of the crate which has the output connectors (Hot Links). The FIC in this crate will accept inputs from the four L2CFTBC's. The FIC will then output the same data it received from each of the four L2CFTBC's to four of the Hot Links connectors on the front of the Magic Bus Transceiver card (described below). The FIC resides in slot 3 of the CTT crate with the FIC backplane card directly behind it. Although this card uses an MBus connector to communicate with the backplane card, it is not actually on the MBus.
- Magic Bus Transceiver card (MBT, see Ref. [5]): This board provides the I/O link between the L2CTT and three outside devices: 1) The L2CFTBC (via the FIC), 2) The L2G and 3) The Serial Command Link (SCL) Hub[2]. The MBT then communicates with the two Alpha processor cards over the MBus (the Alpha cards are described below). Currently it allows input from 8 sources (outside the crate), one of which is used to communicate with the SCL and 4 of which are from the 4 L2CFTBC's (via the FIC). The MBT resides in slot 7.
- Worker and Administrator (see Ref. [8]): These functions are implemented using a general purpose computer (the CPU is a DEC Alpha 21164) on a custom motherboard (based on the PC164 design) with VME and Mbus connections. Both Alpha cards have the same hardware and so only software distinguishes the two. The Administrator does the housekeeping tasks (i.e, manages the input and output event buffers, coordinates

[2]For a brief discussion of the SCL see Appendix A. For a more detailed discussion see Ref. [6].

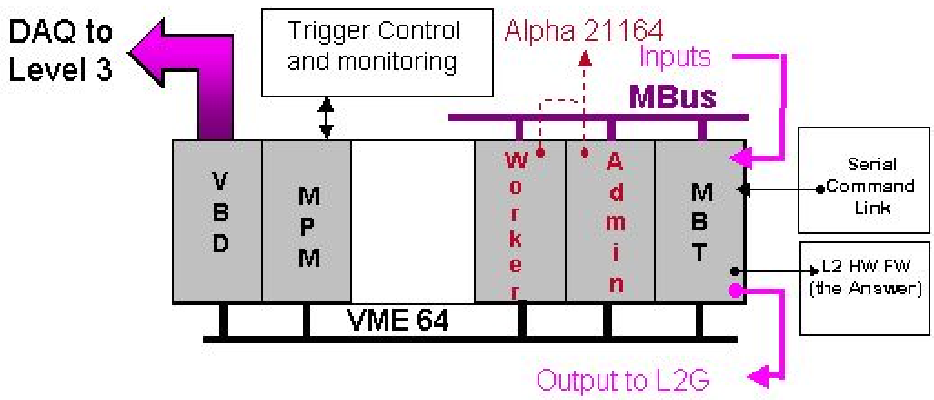

Figure 2: This is schematic drawing of the L2CTT crate. Note that the FIC's are not shown. For a more specific mapping of which board goes in which slot see Table 1.

L3 readout and sends monitoring information to the Trigger Control Computer (TCC)) while the Worker does the track processing. The Worker Alpha resides in slot 10, the Administrator in slot 20.

- Bit3 card (see Ref. [9]): This card performs a few different functions. First, it controls the VMEbus. Second, it allows a computer (a PC) to access the VMEbus. In particular this board provides the interface between the VMEbus and the Trigger Control Computer (TCC). The TCC handles run control, downloads run specific information, and collects monitoring information. Lastly, there is a multi-port memory card on the Bit3 card that is accesible over VME. The Bit3 is not on the MBus, and must reside in slot 1 in order to control the VMEbus.

- VME Buffer/Driver (VBD, see Ref. [10]): This board provides the readout to L3 and resides in slot 2. It receives the readout data from the Administrator via the VMEbus.

- Serial Fan-Out (SFO): The SFO is an optional board. When it is being used it will receive input from the FIC and send output to the MBT, rather than the the FIC and MBT being connected directly. The purpose of this board is that it can output to multiple MBT's. This means we could have the SFO connected to two MBT's: one an online L2CTT MBT, and the other a MBT in a testing crate. this would allow testing of a L2CTT crate without disrupting the online crate.

Slot	Card
1	Bit3 Multi-port Memory (Bit3)
2	VBD (readout to L3)
3	Fiber Input Converter (FIC)
4	Spare
5	Serial Fan-Out (SFO)
6	Spare
7	Magic Bus Transceiver (MBT)
8-9	Spare
10-11	L2 Worker Alpha
12-19	Spare
20-21	L2 Administrator Alpha

Table 1: The cards in the L2CTT crate.

4 Inputs to L2CTT

The L2CTT takes inputs from multiple sources. The primary inputs are from:

- L1CFT
- TCC
- SCL

4.1 Non-L1CFT Inputs

The communication with all inputs other than the L1CFT are described in more detail in Refs. [1, 5, 6] and references therein. Briefly, the Trigger Control Computer (TCC) monitors triggering, luminosity, and other things such as temperature. The TCC also helps in configuring hardware including such tasks as programming gate arrays. The TCC uses the Bit3 card to communicate with the crate. Essentially, the job of the SCL is to coordinate the activities of the various crates. Such things as signaling a crate to readout to L3 is done via the SCL. For a description of the SCL system see Appendix A. The SCL interactions are made through the MBT.

4.2 Inputs from L1CFT

The data transfer between the four L2CFTBC and the L2CTT is implemented using G-links at 53 MHz to send 16-bit words which are read into a FIC. The link activity is as follows: Idle...Data-for-one-event...Idle...Data-for-one-event... where Idle is a G-link defined state in which no data is sent and empty frames are sent to fill the entire time between data transmissions. Note that no information is needed from the SCL as the G-link on the L2CFTBC will send a signal to the FIC telling it that data is coming. Each L2CFTBC will pad the Data-for-one-event to be a multiple of 4 longwords (4×32 bits = 128 bits).

The data is passed through the FIC and on to the MBT via 10-bit transfers with 8 bits of data. Within the MBT the 8-bit words are then converted into 128-bit blocks for passing to the Worker Alpha. Additionally, the FIC adds on a 16 data bit trailer at the end of the transfer. Then the MBT pads this with 112 bits after the FIC trailer so that the transfer size remains a multiple of 128 bits. The data format, as passed to the worker, for each event consists of up to 51 longwords (per L2CFTBC card) of information with the usual 32-bits per longword as shown in Table 2. The data consist of 4 header longwords followed by up to 46 longwords of track data followed by one End-of-Record longword. The data are constructed such that each longword contains information about one and only one track. A maximum of 816 bytes of data (4 boards $\times$ 51 longwords/board $\times$ 4 bytes/longword) is passed for each event. Each event is padded by the L1CFT to be a multiple of four longwords. As mentioned above, the FIC and MBT will add on an extra four longwords to the end.

The bit construction of the header and trailer information sent from the L2CFTBC's are given in Tables 3 and 4. This format is identical to that sent to L2G with a few exceptions which are specified in Table 5. Header 1 contains overview information about the event including the number of tracks and headers sent. Header 2 contains information about

Header Longword 1
Header Longword 2
Header Longword 3
Header Longword 4
Data Longword 0 (Track 0)
⋮
Data Longword N (Track N)
End-of-Record Longword

Table 2: The data longwords sent from the L2CFTBC to the L2CTT. Note that in this example there are N tracks, where N is an integer less than or equal to 46. We have ignored the extra End-of-Record longwords which will be sent to pad the event such that the total data sent is a multiple of 4 longwords. The format from L2CFTBC is 16 data bits per frame, and from FIC to MBT is 8 data bits per frame.

which event is being studied. Header 3 contains processing information including input status bits (defined in Table 6). Header 4 cotains CFT-specific information and indicates the number of tracks in each P_T threshold bin.

Table 7 gives the bit assignments for the data longwords which contain tracks. Within the data longwords bits 0-3 contain preshower and isolation information which is not used by the L2CTT but is passed along to the L2G. Bits 8 and 9 contain information on how much energy was deposited in the CPS. Bits 10-15 contain the six bits of P_T (including the sign) from the CFT . Bits 20-22 contain the octant number (varies from 0-7). The octant number just represents which ϕ-octant the track is in. Bits 16-19 contain the sector offset (which varies from 0 to 9) which tells which sector within the octant contains the track. Since there are 80 sectors and 8 octants there are then 10 sectors per octant. The sector offset just tells which of these 10 sectors in the octant contains the track. Bits 26-31 contain the H layer fiber number within the sector which varies from 0-43. The sector number and the H layer fiber number are used to construct ϕ.

bit #	Header 1	Header 2	Header 3
31	Header Format(2)	Rotation #(15)	Status Bits(7)
30	Header Format(1)	Rotation #(14)	Status Bits(6)
29	Header Format(0)	Rotation #(13)	Status Bits(5)
28	Object Format(4)	Rotation #(12)	Status Bits(4)
27	Object Format(3)	Rotation #(11)	Status Bits(3)
26	Object Format(2)	Rotation #(10)	Status Bits(2)
25	Object Format(1)	Rotation #(9)	Status Bits(1)
24	Object Format(0)	Rotation #(8)	Status Bits(0)
23	Object Length(7)	Rotation #(7)	Processing Bits(7)
22	Object Length(6)	Rotation #(6)	Processing Bits(6)
21	Object Length(5)	Rotation #(5)	Processing Bits(5)
20	Object Length(4)	Rotation #(4)	Processing Bits(4)
19	Object Length(3)	Rotation #(3)	Processing Bits(3)
18	Object Length(2)	Rotation #(2)	Processing Bits(2)
17	Object Length(1)	Rotation #(1)	Processing Bits(1)
16	Object Length(0)	Rotation #(0)	Processing Bits(0)
15	Header Length(7)	Bunch #(7)	Algorithm Minor Version(7)
14	Header Length(6)	Bunch #(6)	Algorithm Minor Version(6)
13	Header Length(5)	Bunch #(5)	Algorithm Minor Version(5)
12	Header Length(4)	Bunch #(4)	Algorithm Minor Version(4)
11	Header Length(3)	Bunch #(3)	Algorithm Minor Version(3)
10	Header Length(2)	Bunch #(2)	Algorithm Minor Version(2)
9	Header Length(1)	Bunch #(1)	Algorithm Minor Version(1)
8	Header Length(0)	Bunch #(0)	Algorithm Minor Version(0)
7	Number of Objects(7)	Data Type(7)	Algorithm Major Version(7)
6	Number of Objects(6)	Data Type(6)	Algorithm Major Version(6)
5	Number of Objects(5)	Data Type(5)	Algorithm Major Version(5)
4	Number of Objects(4)	Data Type(4)	Algorithm Major Version(4)
3	Number of Objects(3)	Data Type(3)	Algorithm Major Version(3)
2	Number of Objects(2)	Data Type(2)	Algorithm Major Version(2)
1	Number of Objects(1)	Data Type(1)	Algorithm Major Version(1)
0	Number of Objects(0)	Data Type(0)	Algorithm Major Version(0)

Table 3: The bit pattern for the first three header words in the data transfer format. The format of the Header and Trailer sent to the L2CTT from the L2CFTBC's and from the L2CTT to the L2G are identical except for the differences listed in Table 5. The Number of Objects is the number of tracks. Header Length, Object length and Data Type are as specified in Table 5. Bunch # is the 8-bit crossing number and Rotation # refers to the turn number of the bunches in the accelerator and is a 16-bit value. These two quantities are generated at the FE boards and are used for event alignment. The Algorithm Version refer to the software algorithms, the Processing Bits refer to the hardware configuration version. Status is as defined in Table 6.

bit #	Header 4	End-of-Record
31	Positive Tracks in P_T Bin 3 (0=No/1=Yes)	Longitudinal Parity(15)
30	Negative Tracks in P_T Bin 3 (0=No/1=Yes)	Longitudinal Parity(14)
29	# Tracks in P_T Bin 3(5)	Longitudinal Parity(13)
28	# Tracks in P_T Bin 3(4)	Longitudinal Parity(12)
27	# Tracks in P_T Bin 3(3)	Longitudinal Parity(11)
26	# Tracks in P_T Bin 3(2)	Longitudinal Parity(10)
25	# Tracks in P_T Bin 3(1)	Longitudinal Parity(9)
24	# Tracks in P_T Bin 3(0)	Longitudinal Parity(8)
23	Positive Tracks in P_T Bin 2 (0=No/1=Yes)	Longitudinal Parity(7)
22	Negative Tracks in P_T Bin 2 (0=No/1=Yes)	Longitudinal Parity(6)
21	# Tracks in P_T Bin 2(5)	Longitudinal Parity(5)
20	# Tracks in P_T Bin 2(4)	Longitudinal Parity(4)
19	# Tracks in P_T Bin 2(3)	Longitudinal Parity(3)
18	# Tracks in P_T Bin 2(2)	Longitudinal Parity(2)
17	# Tracks in P_T Bin 2(1)	Longitudinal Parity(1)
16	# Tracks in P_T Bin 2(0)	Longitudinal Parity(0)
15	Positive Tracks in P_T Bin 1 (0=No/1=Yes)	Data Type(7)
14	Negative Tracks in P_T Bin 1 (0=No/1=Yes)	Data Type(6)
13	# Tracks in P_T Bin 1(5)	Data Type(5)
12	# Tracks in P_T Bin 1(4)	Data Type(4)
11	# Tracks in P_T Bin 1(3)	Data Type(3)
10	# Tracks in P_T Bin 1(2)	Data Type(2)
9	# Tracks in P_T Bin 1(1)	Data Type(1)
8	# Tracks in P_T Bin 1(0)	Data Type(0)
7	Positive Tracks in P_T Bin 0 (0=No/1=Yes)	Bunch #(7)
6	Negative Tracks in P_T Bin 0 (0=No/1=Yes)	Bunch #(6)
5	# Tracks in P_T Bin 0(5)	Bunch #(5)
4	# Tracks in P_T Bin 0(4)	Bunch #(4)
3	# Tracks in P_T Bin 0(3)	Bunch #(3)
2	# Tracks in P_T Bin 0(2)	Bunch #(2)
1	# Tracks in P_T Bin 0(1)	Bunch #(1)
0	# Tracks in P_T Bin 0(0)	Bunch #(0)

Table 4: The bit pattern for the fourth header word and the trailer word. Note that while the trailer word is identical both into and out of the L2CTT, the fourth header word is for use into the L2CTT and does not get sent to L2G. Bunch Number and Data Type are the same as specified in Tables 3 and 5 respectively. In header 4, for each bin, information is passed as to whether or not there are one or more tracks with a given sign in the bin. This information is passed in bits 6 & 7, 14 & 15, 22 & 23, and 30 & 31.

Header/Trailer Word	Values into L2CTT	Values out to L2G/L3
Header Length	4 (in longwords)	3 (in longwords)
Object Length	1 (in longwords)	2 (in longwords)
Object Format	0	0
Header Format	1	1
Data Type	175 (L2CFTBC 0) 176 (L2CFTBC 1) 177 (L2CFTBC 2) 178 (L2CFTBC 3)	5
Algorithm Major Version	1	1
Algorithm Minor Version	1	1
Processing Bits	0	0
Status Bits	See Table 6	

Table 5: The definition of the header and trailer words sent to L2CTT and from L2CTT. See Tables 3 and 4. Note that only the first three header words are sent to L2G.

Value	Meaning
7	Any kind of error during processing; Use at own risk
6	No processing Attempted (none required)
5	Object list truncated (any reason)
4	Receiver error on some input physical trailer
3	Spare
2	Spare
1	More data-type info: 0=MC, 1=Min-bias
0	Real Data (1 = Not Normal, 0=Normal)

Table 6: This table specifies the meaning of the Status Bits word in Header 2. We note for completeness that the Object list truncated value is never 1 in the value sent to L3 from the L1CFT.

bit #	Data Word
31	H Layer Fiber Number(5)
30	H Layer Fiber Number(4)
29	H Layer Fiber Number(3)
28	H Layer Fiber Number(2)
27	H Layer Fiber Number(1)
26	H Layer Fiber Number(0)
25	Isolated CFT Track (0=No/1=Yes)
24	Isolated CFT/CPS (0=No/1=Yes)
23	L2STT Duplicate Track
22	Octant Number(2)
21	Octant Number(1)
20	Octant Number(0)
19	Sector Offset(3)
18	Sector Offset(2)
17	Sector Offset(1)
16	Sector Offset(0)
15	Sign of the P_T for Track
14	P_TBin(1)
13	P_TBin(0)
12	Extended P_TValue(2)
11	Extended P_TValue(1)
10	Extended P_TValue(0)
9	Match to High Threshold CPS (0=No/1=Yes)
8	Match to Low Threshold CPS (0=No/1=Yes)
7	ERROR Code(2)
6	ERROR Code(1)
5	ERROR Code(0)
4	Preshower Cluster Sector (0=Same Sector/1=Neighbor Sector)
3	Preshower Cluster Relative Address(3)
2	Preshower Cluster Relative Address(2)
1	Preshower Cluster Relative Address(1)
0	Preshower Cluster Relative Address(0)

Table 7: The bit assignments for the data longwords containing tracks from the L2CFTBC to the L2CTT. Within the data longwords bits 0-3 contain preshower and isolation information which is not used by the L2CTT but is passed along the L2G. Bits 10-15 contain the three bits of P_T including the sign. Bits 20-22 contain the octant number which varies from 0-7. Bits 16-19 indicate which sector within that octant and varies from 0-9. Bits 26-31 contain the H layer fiber number within the sector which varies from 0-43.

5 Outputs of L2CTT

The outputs of the L2CTT go to:

- L2G
- L3
- TCC
- SCL

5.1 Non-L2G Output

The output to devices other than the L2G are described in more detail in Refs. [1, 5, 11] and references therein. To summarize, stored input and output values of events which are accepted by L2 and sent to L3 are implemented using standard techniques with Administrator Alpha passing information via the VBD. The TCC interactions use the Bit3. The SCL interactions are through the MBT. See Appendix A for more details about the SCL connections and the information passed, and Section 7 for SCL interactions on errors.

5.2 Output to Level 2 Global

The data transfer between the L2CTT and L2G is implemented using the standard MBT-to-MBT connection which sends the sorted track objects out of the front panel of the MBT to the "Pilot" MBT in the L2G crate via a Cypress Hotlinks cable [5]. The data format begins with the standard header followed by the track objects and eventually followed by the standard trailer.

The data format for each event consists of up to 372 longwords of information with 32-bits per longword as shown in Table 8. Each transmission consist of 3 header longwords followed by up to 368 longwords of track data (4 L2CFTBC $\times$ 46 tracks (maximum)/concentrator $\times$ 2 words per track) followed by an End-of-Record longword. Each track requires two data longwords. A maximum of 1,488 bytes of data (372 longwords $\times$ 4 bytes/longword) is passed for each event. Each event is padded (using a repeated footer) to be a multiple of 4 longwords[3].

The format for the header and trailers from L2CTT to L2G is the standard preprocessor to L2G format [1]. The requirements of the data transfer are given in more detail in Appendix B. The bit configuration is given in Tables 3 and 4. The values for the header and trailer words which are sent to the L2CTT and those which are sent from L2CTT to L2G (and L3) are given in Table 5. The headers contain overall event information, algorithm and transmission information. Header 3 contains the total number of tracks being sent.

Table 9 gives the bit assignments for the data longwords which contain tracks. Data Word 0 contains the extrapolated ϕ position at EM3 (ϕ_{EM3}) and the $\mathrm{P_T}$. Word 1 contains

[3]Note that there is a maximum of 816 bytes coming into the L2CTT from the L1CFT. The difference is due to the fact that L2CTT receives one longword per track from the L1CFT but then transmits two longwords per track to the L2G (also, there are differences in formatting).

Header Longword 1
Header Longword 2
Header Longword 3
Data Longword 0 (Track 0, Word 0)
Data Longword 1 (Track 0, Word 1)
Data Longword 2 (Track 1, Word 0)
Data Longword 3 (Track 1, Word 1)
⋮
Data Longword 2N+1 (Track N, Word 1)
End-of-Record

Table 8: The data longwords sent from the L2CTT to L2G. Note that in this example there are N tracks, where N is an integer less than or equal to 184.

the ϕ_0 for the track, the isolation and preshower information passed from the L1CFT and L2 isolation.

bit #	Data Word 0	Data Word 1
31	$P_T(15)$	Spare2(15)
30	$P_T(14)$	Spare2(14)
29	$P_T(13)$	Spare2(13)
28	$P_T(12)$	Spare2(12)
27	$P_T(11)$	Spare2(11)
26	$P_T(10)$	Spare2(10)
25	$P_T(9)$	Spare2(9)
24	$P_T(8)$	Spare2(8)
23	$P_T(7)$	Spare2(7)
22	$P_T(6)$	Spare2(6)
21	$P_T(5)$	Spare2(5)
20	$P_T(4)$	Spare2(4)
19	$P_T(3)$	Spare2(3)
18	$P_T(2)$	Spare2(2)
17	$P_T(1)$	Spare2(1)
16	$P_T(0)$	Spare2(0)
15	$\phi_{EM3}(7)$	High Thresh CPS Match
14	$\phi_{EM3}(6)$	Low Thresh CPS Match
13	$\phi_{EM3}(5)$	Isolated CFT/CPS
12	$\phi_{EM3}(4)$	Isolated CFT
11	$\phi_{EM3}(3)$	L2iso(1)
10	$\phi_{EM3}(2)$	L2iso(0)
9	$\phi_{EM3}(1)$	Spare3(0)
8	$\phi_{EM3}(0)$	Sign
7	Spare1(7)	$\phi_0(7)$
6	Spare1(6)	$\phi_0(6)$
5	Spare1(5)	$\phi_0(5)$
4	Spare1(4)	$\phi_0(4)$
3	Spare1(3)	$\phi_0(3)$
2	Spare1(2)	$\phi_0(2)$
1	Spare1(1)	$\phi_0(1)$
0	Spare1(0)	$\phi_0(0)$

Table 9: This table gives the specification of the track objects output to L2G. The maximum amount of data sent is $\sim$ 185 obj/evt $\rightarrow$ 1,500B/evt

6 Internal Functioning of the L2CTT

6.1 The FIC to MBT Connection

As previously mentioned, the results from the four L2CFTBCs are read in via a FIC board. The FIC converts the input into the proper format and passes its results out a front panel connector [4] to the MBT via HotLinks. Within the MBT [5] the data are fed into FIFO memories and the end of an event is marked in the FIFO by a control signal. The FIFO can hold 16 events[4].

The data transfer from the MBT to the Administrator and Worker is negotiated between the Administrator and the MBT. The control section of the MBT knows which FIFOs are active and notes when a full event is available in each input FIFO. Data are not transferred out on the MBus until a full event is available. A GO control message from the Administrator signals the MBT that it is ready for event transfer and transmission begins when the MBT cards has an event ready to send. When all FIFO's have sent their event a "DONE" MBus signal is sent.

6.2 Calculation Algorithms

The primary functionality of the L2CTT processing occurs on the Worker Alpha which contains a CPU running C++ code. The processing begins on the Worker Alpha after the Administrator has indicated to the Worker that it may process the next event. The following sections describe the sections of the code which calculate the $\mathrm{P_T}$ ϕ_0, the extrapolated track position ϕ_{em3} and isolation properties of each track. In addition to L2 isolation (called cttL2iso) there is an L1 isolation variable which is simply passed from L1 to L2G without any change.

6.2.1 P_T calculation

At L2CTT preprocessor $\mathrm{P_T}$ is calculated from information provided by L1. A first approximation of $\mathrm{P_T}$ is given by $\mathrm{P_T}$ bin. Each value of $\mathrm{P_T}$ bin corresponds to a particular momentum range: bin 0 for $\mathrm{P_T} > 10$ GeV, bin 1 for $5 < \mathrm{P_T} \leq 10$ GeV, bin 2 for $3 < \mathrm{P_T} \leq 5$ GeV and bin 3 for $1.5 < \mathrm{P_T} \leq 3$ GeV. A better (and final) approximation of $\mathrm{P_T}$ can be obtained from extended $\mathrm{P_T}$ which is also provided by L1. The extended $\mathrm{P_T}$ associated with bins 0, 1 or 2 is an integer ranging from 0 to 3. For bin 3 extended $\mathrm{P_T}$ ranges from 0 to 7. For high momentum bins (0 and 1), extended $\mathrm{P_T}$ just represents the $\mathrm{P_T}$ sub-bin. Therefore, it is straightforward to associate a $\mathrm{P_T}$ value (which will be sent to L2G) for each pair of bin and sub-bin. This information is depicted in Table 10.

For momentum lower than 5 GeV (bins 2 and 3) a better momentum resolution can be achieved if the momentum is evaluated directly from the track curvature. In order to determine the momentum, we need to know at least 3 points (in $r-\phi$ plane) consistent with track trajectory. One of these points is the origin, which is a good approximation for the starting point of the track. The second point is determined using the H-layer fiber which

[4] The FIFOs are actually 64KB/channel. But this can easily accommodate 16 events worth of raw tracking data as 16 events at 816 bytes/Event is only 13,056 bytes.

Bin #	P_T Bin ID & Sub-Bin ID	P_T^{Min} (GeV/c)	P_T^{Max} (GeV/c)	P_T^{Mean} (GeV/c)
0	00000	39.55	inf	77.95
0	00001	19.93	39.55	26.50
0	00010	13.32	19.93	15.96
0	00011	10.00	13.32	11.42
1	01000	8.00	10.00	8.89
1	01001	6.67	8.00	7.27
1	01010	5.71	6.67	6.15
1	01011	5.00	5.71	5.33

Table 10: The conversion between the 5-bits of P_T information reported by L1CFT and the P_T reported to L2G for the "H and P_T" method. Remember the two most significant bits of P_T BinID represent the Bin number. The number that will be reported to the L2G is the P_T^{Mean}.

is hit by the charged particle. This information is available at L2CTT because the sector number and fiber H number within the sector are known from L1. To have a third point, we should have similar information about another CFT layer (preferably the A layer). Sending the A fiber number to L2 requires too many bits, therefore only the offset of the hit fiber in layer A is available at L2. The offset definition derives from geometrical considerations. Assume that a straight line passing through the hit fiber H and the origin crosses a fiber in layer A. Designate this fiber as "i". Also assume that the particle's trajectory crosses a fiber in layer A. Designate this fiber as "j". By definition the offset is $\mid i - j \mid$. For each track with P_T bin 2 or 3, the value of the offset is sent from L1 to L2 as extended P_T. So, the extended P_T is sub-bin number for bins 2 and 3 and proportional to the offset for bins 0 and 1. More details about the P_T calculation with the offset method can be found in Appendix C.

Momentum resolution for different momentum ranges was estimated by plotting the difference between the trigger momentum and the generated momentum. Studies were performed for various Monte Carlo samples and the momentum resolutions were consistent. The momentum resolution for $Z \rightarrow \tau\tau + 4$ min bias is shown in Fig. 3

6.2.2 ϕ_0 and ϕ_{em3} calculations

The calculation of the angular variables ϕ_0 and ϕ_{em3} relies exclusively on geometrical considerations. With P_T and ϕ_H known (from fiber H number and sector number) the value of ϕ_0 is given by:

$$\sin(\phi_H - \phi_0) = \frac{0.15 B R_H}{P_T} \tag{1}$$

where B is the intensity of the magnetic field (2 T) and R_H is the radius of H layer (0.5152 m). The previous equation can be solved for ϕ_0 by approximating ϕ_0 - sin in order to minimize the computational time.

Figure 4 shows the ϕ_0 resolution for a Monte Carlo sample consisting of single electron events. The ϕ_0 resolution is dominated by two sources: the P_T resolution and the limited number of bits available for storing ϕ_0. The value for ϕ_0 to be sent to L2G must fit into 8

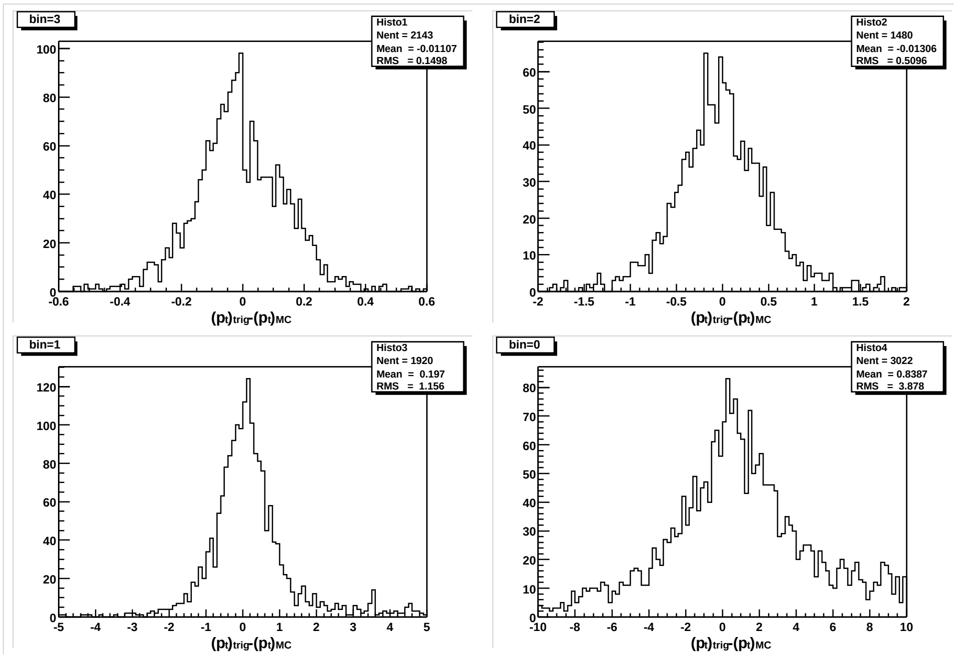

Figure 3: Momentum resolution for $Z \to \tau\tau + 4$ min bias events. Only π, μ and e tracks originating from a τ were taken into account.

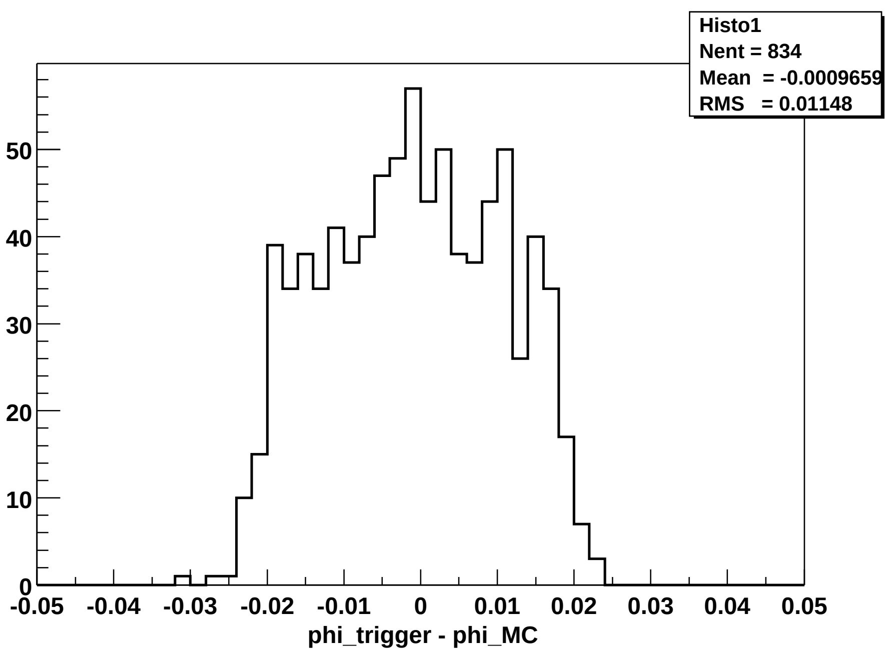

Figure 4: ϕ_0 resolution for single electron events.

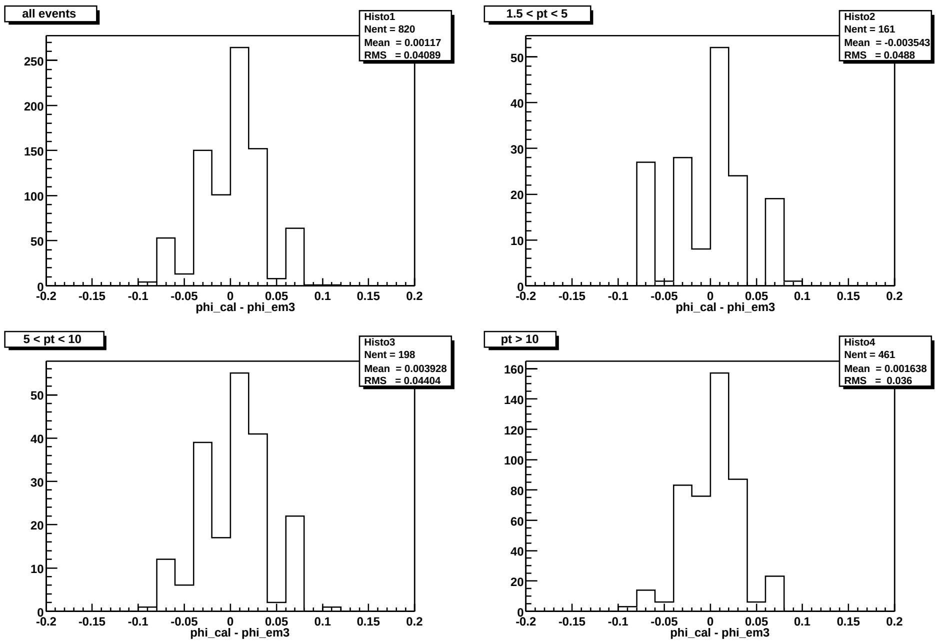

Figure 5: $\phi_{em3} - \phi_{tower}$ for single electron events.

bits so the resolution could be $\frac{256}{2\pi}$. However, we only use a resolution of $\frac{160}{2\pi}$. So to convert ϕ_0 to an 8 bit number to send to L2G, we take $Round[\phi_0 \cdot \frac{160}{2\pi}]$. A similar process is used for conversion of $\mathrm{P_T}$ to an 8-bit number: the formula $Round[4 \cdot P_T]$. In this case the value of 64 would be sent as $\mathrm{P_T}$.

To extrapolate ϕ_0 to the third calorimeter layer (ϕ_{em3}) we assume a circular path from the origin to the center of the solenoid followed by a linear motion to Em_3. This model is simple enough to allow an anlytical expression for ϕ_{em3}:

$$\phi_{em3} \approx \phi_0 \pm \frac{0.15BR_{solenoid}}{P_T}\left(2 - \frac{R_{solenoid}}{R_{em3}}\right) \qquad (2)$$

The + sign should be chosen for positive tracks.

The main purpose for estimating ϕ_{em3} is to have a matching criterion between tracks found in the CFT and those hit towers in the EM calorimeter. To estimate the validity of our model, the calculated values of ϕ_{em3} were compared with the angles corresponding to the hit calorimeter towers. Since the angular width of a tower is $2\pi/32$ (≈ 0.2) a resolution of about 0.04 is expected. Most of the tracks leave energy in two adjacent towers and therefore the resolution is a little better than what we normally expect ($0.2/\sqrt{12} \approx 0.058$). Figure 5 shows the good agreement between the two angles.

6.2.3 L2 Isolation

An algorithm for finding isolated tracks was implemented in order to enhance trigger capabilities on τ leptons [3]. An isolated track must have $\mathrm{P_T} > 5$ GeV and can be either 1-prong (no other track with $\mathrm{P_T} > 2$ GeV within 25 ϕ bins and at most two tracks with any $\mathrm{P_T}$ within

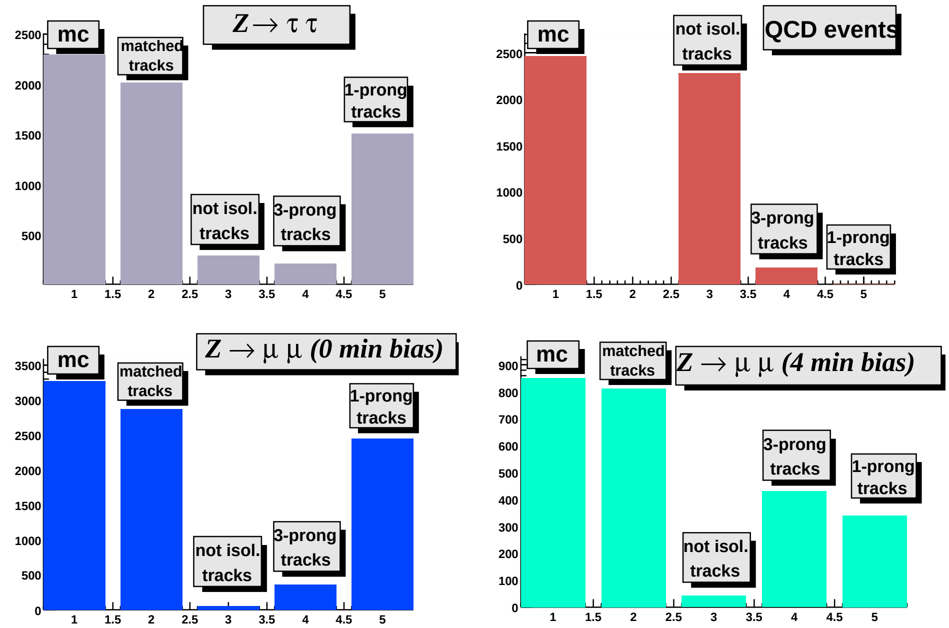

Figure 6: Efficiency of finding isolated tracks for different Monte Carlo samples. The first bin represent generated Monte Carlo tracks within the CFT acceptance. The second bin represents the trigger tracks matched with input values of ϕ_0.

25 ϕ bins) or 3-prong (at most 4 tracks with any $\mathrm{P_T}$ within 2 ϕ bins). If a track is 3-prong isolated, the variable cttL2iso is incremented with 2 and if the track is 1-prong, cttL2iso is incremented with 1. So, the variable cttL2iso can be 3 (isolated track), 2 (isolated 3-prong), 1 (isolated 1-prong) or 0 (not isolated track). The efficiency of finding isolated tracks can be seen in Fig. 6 for different Monte Carlo samples. QCD events (the main background) are heavily rejected by the isolation criterion.

6.3 Sorting Algorithm

We use optimized bubble sort algorithm in the Worker Alpha to order the tracks by $\mathrm{P_T}$ for output to L2G. Specifically, we create a list of pointers. Each pointer points to a track object.

The algorithm begins by concatenating the tracks from the different L2CFTBC boards. For each track we evaluate $\mathrm{P_T}$ using either the look up table (bins 0 and 1) or the offset method (bins 2 and 3). Although it is in general true that tracks in bin i have higher $\mathrm{P_T}$ than in bin $i+1$, there are situations when this statement is not true. For example tracks in bin 2 may have $\mathrm{P_T}$ higher than some of the tracks in bin 1, if their offset is low. Therefore we do not assume any $\mathrm{P_T}$ bin ordering.

Once the $\mathrm{P_T}$ for each track is in place, the algorithm is as follows: we take the pointer on the second track of the list (assuming there is a track as indicated using Header 3 from each of the L2CFTBC boards) and compare its $\mathrm{P_T}$ content with the $\mathrm{P_T}$ of the prevoius (the first) pointer. If the first track's $\mathrm{P_T}$ is larger than the second one, pointers are exchanged. In general, the i-th pointer is exchanged with the first j-th pointer ($j < i$) which P_T is smaller than the P_T of the i-th one. We repeat until all the pointers from the list are used. In this method the lowest P_T tracks fall to the bottom.

6.4 L2 Object Outputting

After all the tracks have been sorted, the L2 objects are created using the format in Table 9. These track objects are then copied, by P_T order, to the output memory location for output to L2G. The output location is specified (passed by reference to the L2CTT object as noted above) by the data movement control section of the Worker as a reference to the beginning of an array.

6.5 Scalar and Monitoring information

Here we specify the list of scalar and monitoring information which will be readout as well as its output format. For now we list the things we need to store: the number of events, errors, total number of tracks seen, tracks per event, number of truncated events, number of saturated events and, the number of synchronization errors. We note that some of these will be stored on the Worker while others will be collected on the Administrator.

7 Errors and Error Handling

There are a number of error conditions which need to be identified by the L2CTT. In addition there are a number of error condition to which the L2CTT has to respond. At the moment the only one which has been discussed is the case of events with mismatched pieces.

The Administrator Alpha looks at all pieces of the event to check if all the pieces are indeed from the same event. If they are not, this causes an error, and the Administrator sends this information to the MBT by writing to a register on the MBT. The SCL Mezzanine card (on the MBT) then notifies the L2 HWFW of the synchronization error (L2ERROR). The Administrator keeps count of these occurrences.

The SCL Hub responds by broadcasting *SCL Initialize* to which MBT responds by raising L1_BUSY. Since any preprocessor node can invoke a *SCL Initialize*, Administrator must be notified by MBT. After notification from MBT, the Administrator clears its buffers and requests that the Workers do the same. After clearing their buffers the Workers notify the Administrator. After all Workers have notified the Administrator, the Administrator notifies the MBT to rescind the L1_BUSY.

In the event of all L3 readout buffers being full, or all Wait for L2 Decision Buffers being full, the front end automatically throttles the framework and no more events are sent to L2CTT.

8 Simulations

We have performed a number of simulations designed to test the prototype of the L2CTT worker code. To test the L2CTT object creation and sorting algorithms we have created a set of toy simulations for track generation, the L1CFT reconstruction, and outputs and the L2CTT crate surrounding the Worker Alpha. The simulation uses C++ code. For simplicity the original timing studies are done by compiling and running using Microsoft Visual C++ (Version 6.0) on a 233 MHz Pentium II processor running in a PC running Windows NT. The results were checked using an Alpha running Linux.

The track generator simulation produces events with a very stiff $\mathrm{P_T}$ distribution ($\frac{dN}{d\mathrm{P_T}} \propto e^{-0.1\times \mathrm{P_T}}$) and flat in ϕ_0. For each event a number of tracks are simulated. The L1CFT simulation assumes a perfect detector such that it finds all tracks, has no fakes, and creates L1 track objects, headers and trailers perfectly. There is no smearing, and all information lost is solely due to the resolution of the information transfer protocols. To make conservative timing estimates as well as eliminating any bias due to the $\frac{dN}{d\mathrm{P_T}}$ distribution, each track is placed on their respective L2CFTBC board unsorted by $\mathrm{P_T}$. To simulate the output, each L2CFTBC is given its appropriate headers, data and trailers and outputed into a text file.

The L2CTT simulation mocks up the preprocessor data movement by reading in events from the text file and storing the first 50 events. These events are put into the proper input buffer format, and pointers to the arrays are passed to the Worker prototype code. The Worker algorithm (object creation, sorting and object output) is run 200,000 times on each event and timed. This simulation is repeated for track multiplicities of 0, 1, 2, 4, 8, 16, 32 and 64 tracks for 50 events each. The results are given in Table 11 and shown graphically in Figure 7. For events with less than 40 tracks, the processing time is less than 1 microsecond per track. Note these numbers were checked using an Alpha processor which was found to be a factor of roughly 1.6 faster [5]

To convert this to a processing time distribution we need the $\mathrm{N_{Track}}$ distribution per event into the L2CTT. This is estimated using a separate technique. The Trigger Simulator Ntuples [13] are used to get an estimate of the efficiency and fake rates for finding tracks in the CFT. The trigger simulation takes as input fully simulated dijet events from ISAJET in 6 $\mathrm{P_T}$ bins with varying number of additional min-bias events. These events are then run through a full L1 simulation (i.e, calorimetery, tracking and muon). The simulation does a full track finding and gives an estimate which takes into account the track finding efficiency as well as the fake track rate as a function of the number of extra min-bias collisions in the event. In addition it allows a L1 decision based on a straw L1 trigger table. The number of tracks reported into L2CTT is shown in Figure 8. The different distributions represent samples which have a mean number of interactions per crossing[6] of 0.5 and 5.0. We have taken into account the Poisson nature of the number of actual interactions in an event given the mean number of interactions. In addition the plots show for comparison the case where all events are investigated (no L1 cut or said differently a min-bias sample) and the case in

[5] The timing studies were done with an older version of the code. New timing studies, using the new version of the code, were not done yet.

[6] The mean number of interactions per crossing is a function of the luminosity, the number of bunch crossings and the bunch spacing.

which events are required to pass L1. Each distribution appears to be fairly well modeled by an exponential. Figure 9 shows the mean number of L1CFT track candidates as a function of the mean number of interactions per collisions.

Figure 10 shows the results of combining the processing time per N_{Track} and the N_{Track} per event distributions to show the fraction of events which will be processed within a specified time. From the figure we see that even with an average of 5 interactions per crossing 95% of the events will be processed in less than 60μs. Even for 10 interactions per crossing, we see less than 35 tracks (on average) into L2CTT. As mentioned above, these results should be fairly insensitive to the fact that we used a different P_T spectrum to measure the processing time per event for a given track multiplicity. While these results are clearly not highly accurate as effects such as the noise in the CFT could easily drive the fake track rate higher, we have tried to compensate by not presorting the tracks as done in the L2CFTBC or taking into account the gains from running the code on the Alpha processor (which should be faster than the simulation we have run without optimization). Based on these results we believe the L2CTT, except in extreme situations, should be able to function with minimal deadtime and without additional processing power.

N_{Track}	Make Objects (μSec)	Make and Sort Objects (μSec)	Make, Sort Output (μSec)
0	0.5	0.5	0.8
1	1.1	1.1	1.3
2	1.5	1.5	1.9
4	2.2	2.7	3.1
8	3.6	5.1	5.7
16	6.0	10.8	11.8
32	10.7	25.4	27.4
64	20.7	72.2	78.8

Table 11: The processing time as a function of track multiplicity. Note these numbers were checked using an Alpha processor which was found to be a factor of roughly 1.6 faster. These numbers do not include the data transfer time in and out of the Worker Alpha to L2G.

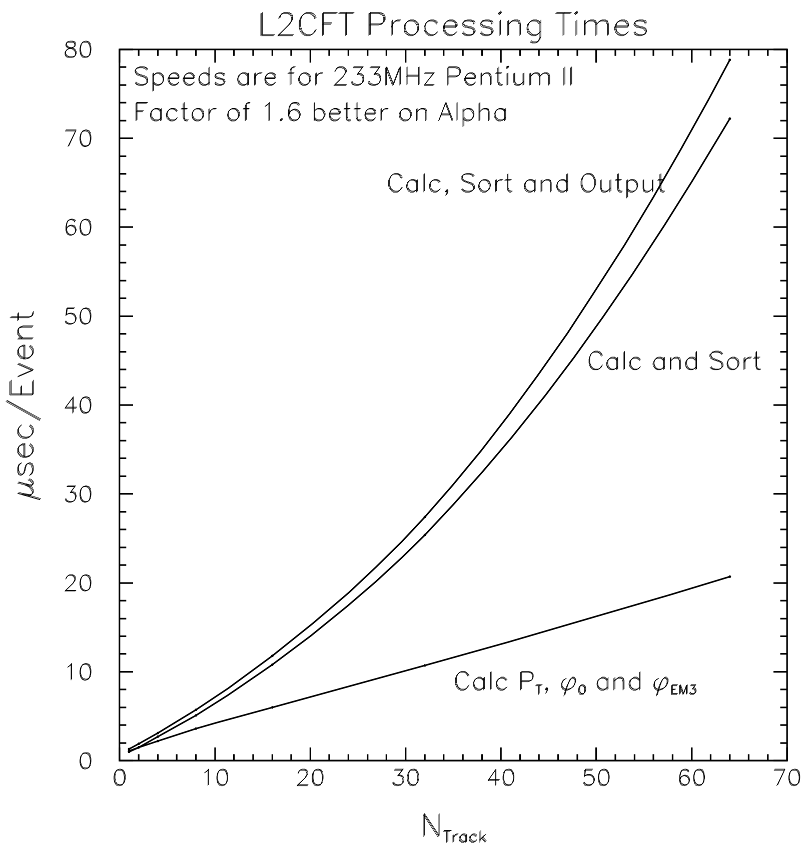

Figure 7: The processing time as a function of the track multiplicity. Note these numbers were checked using an Alpha processor which was found to be a factor of roughly 1.6 faster. These numbers do not include the data transfer time in and out of the Worker Alpha to L2G.

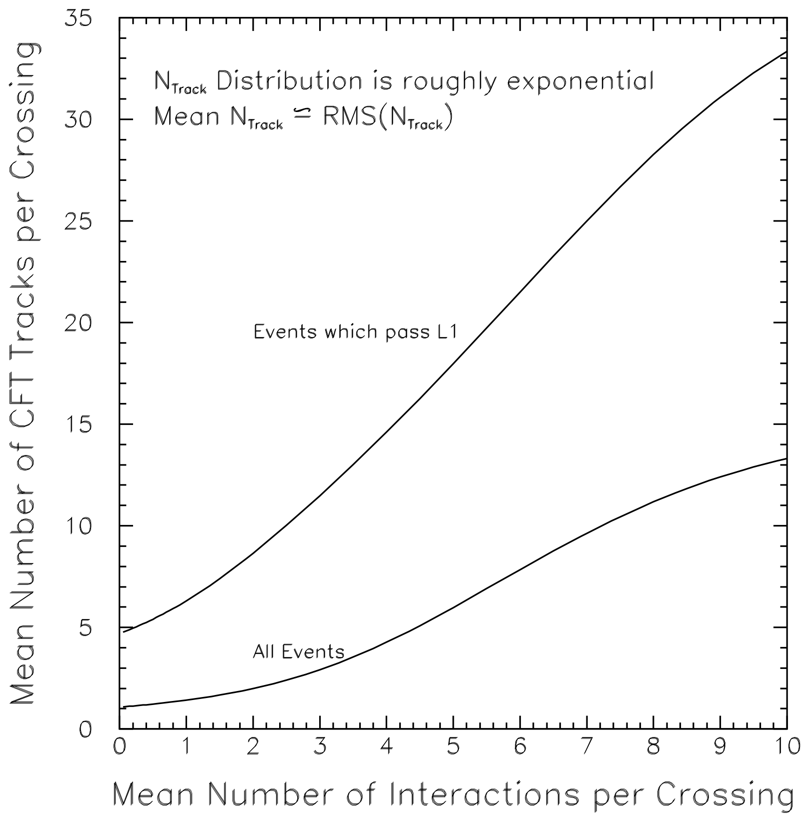

Figure 8: The number of tracks, as reported by the L1CFT, for 0.5 and 5 interactions per crossing using the trigger simulator. Plots show the distributions of the mean number of CFT tracks for all events and for those which pass the straw L1 trigger table. Note that this takes into account the Poisson variation in the number of min-bias events per collision and an estimate of the fake track rate.

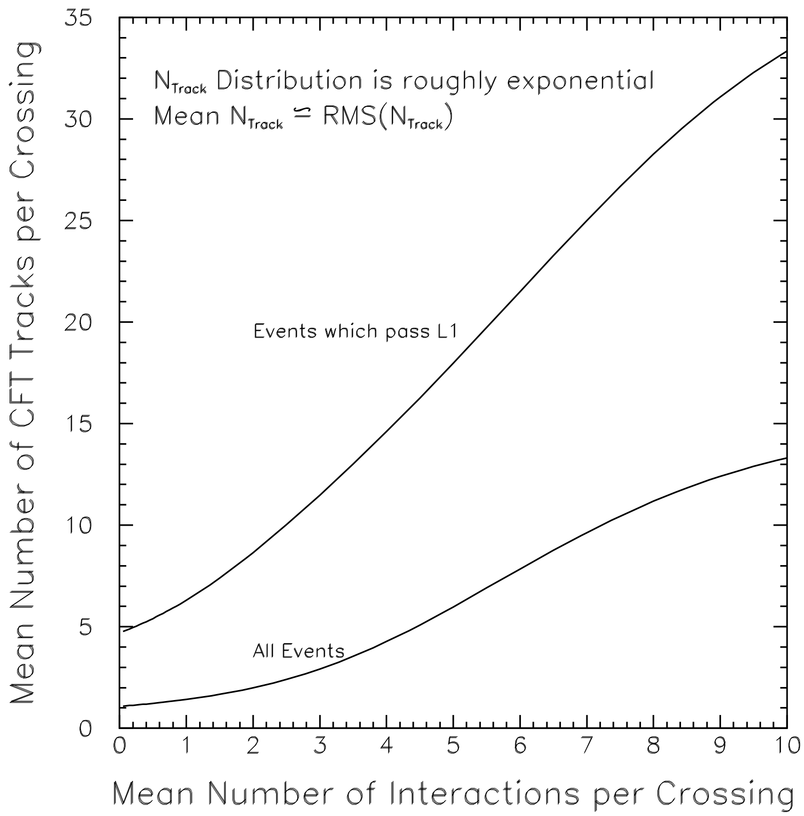

Figure 9: The mean number of tracks, as reported by the L1CFT, as a function of the mean number of events per crossing using the trigger simulator. Note that this takes into account the Poisson variation in the number of min-bias events per collision and an estimate of the fake track rate.

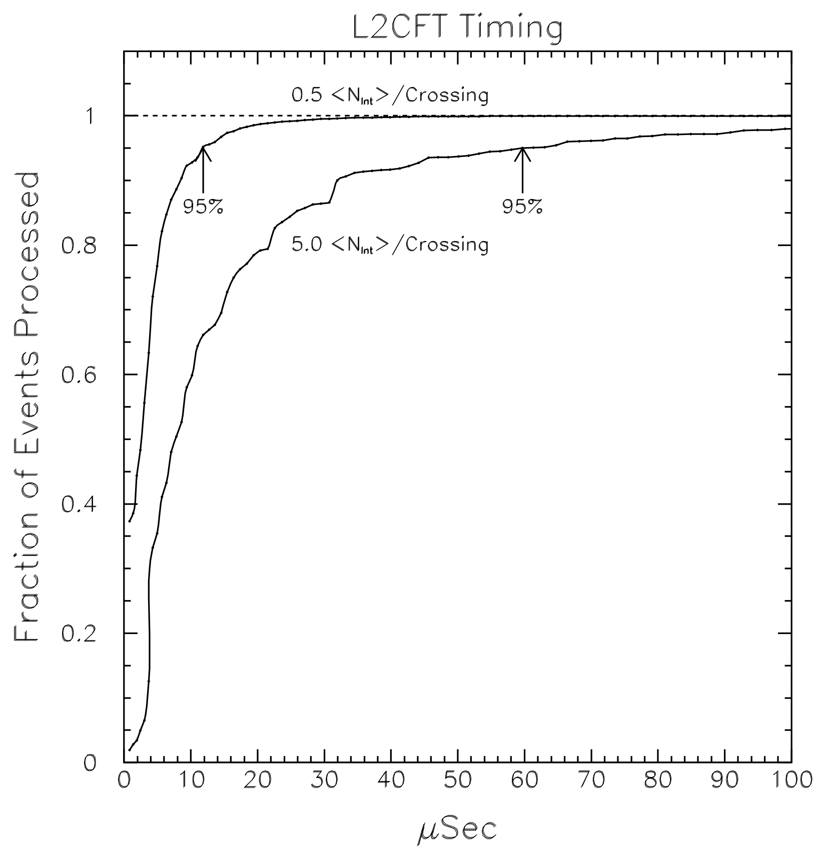

Figure 10: The expected time per event distribution out of the L2CTT. Note that these numbers are conservative in that they use the slower 233 MHz Pentium II processor and have no sorting from L1CFT.

9 High Luminosity Contingency Plans

While studies indicate that there should be enough time to do the full processing and data transfer, there exists the possibility that we may need to cut back on the processing time. This could be the case for running with a large number of tracks sent to L2 per event (e.g. very high luminosity, lots of noise/fakes in the L1CFT etc.). This can be done in a number of ways. For example, we may put a P_T cut requirement to limit the number of tracks sent to L2G. This may mean that we only send information from a certain number of different P_T bins. Another option is to simply limit the number of tracks sent. Either requirement would help insure that we do not get close to the 4kB event size limit (this includes the header and trailer information). A final possibility is to limit the P_T to one Byte and condense the output object from 2 longwords to 1. This possibility would cut down the data size but would not help the L2G processing time.

10 Conclusions

We have presented a technical design report for the Level 2 Central Tracking Trigger preprocessor crate. The inputs and outputs for the crate are fully specified and appear to be adequate for full functionality. A prototype of the Worker code exists and has been used to do proof of principle and timing studies. The results of the timing studies indicate that the crate should be able to operate with minimal dead-time until only the most extreme of circumstances. There are a number of contingency plans already in place to deal with those scenarios. This design is easily expandable to include the L2STT when it comes online.

11 Acknowledgments

The authors would like to sincerely thank Jerry Blazey, Jim Linnemann, Manuel Martin, Rob Martin and Roger Moore for their technical expertise, discussions and careful readings of the text.

References

[1] D. Edmunds, S. Gross, P. Laurens, J. Linnemann and R. Moore *Technical Design Report for the Level 2 Global Processor* DØNote 3402, Feb 1998.

[2] DØ electronics group (Fred Borcherding *et al.*). *Technical Design Report for the Upgrade L1/L2 Tracking Trigger. Version 2* DØ Note XXXX.

[3] Y. Gershtein, D0 TN-3766.

[4] FIC Reference.

[5] D. Baden, *Technical Design Report for the Magic Bus Transceiver Card* DØ Note XXXX.

[6] SCL Reference

[7] Magic bus reference.

[8] Alpha Reference.

[9] Bit3 Reference.

[10] VBD Reference.

[11] L3 Reference

[12] R. Fleischman, Highwind Corp, private communication.

[13] Trigger Simulation Reference

A Serial Command Link Inputs

The L2CTT crate has a direct connection to the SCL which is used to transfer information from the trigger framework (the SCL Hub) to the crate. The SCL has separate input and output connections which connect directly into the MBT via a mezzinine card. The MBT separates the SCL information by source and makes it appear as the 8th data source on the MBT card. For example, the MBT selects the L2 accept/reject SCL messages buffers and makes them available to the Administrator as needed. In this section we summarize some of the SCL commands as they affect the L2CTT. The commands the SCL sends out are given in Table 12.

The SCL sends notification of a L1A to every geographic section which needs to read out to L3. In the case of the L2CTT, the receipt of the L1A SCL message means that at least a header block must be sent to L2G for this event and a header block must be prepared for eventual L3 readout in case the event passes L2.

The MBT often stores information from the SCL. This is done by transforming the input into the correct format and directing it to its input FIFO. Each L1A is accompanied by header/trailer information which is placed in a source FIFO and appears to the rest of the system as a standard data source. The relevant L1 information carried by SCL is the 3 byte crossing number at the time of the L1A and the 2 byte set of L1 Qualifiers.

There are a number of L1 qualifiers. The ones of greatest interest to the L2CTT are the *Unbiased Sample (UBS), Forced Write, L2 Global Needed* and *Collect Status*. The obligation to read out to L3 is announced by receipt of the L1A SCL message, but a separate *L1 Needed* qualifier will be provided to all L2 preprocessors (*L2 Pre_CTT*) and possibly a *L2 Global Needed* may be defined. These qualifiers offer the L2CTT the option to skip execution of its algorithm for an event without its *Needed* qualifier and produce a minimal header-only output to L2G (and preparing a minimal L3 output). This qualifier would be attached to a L1 bit whenever the L2G requires the input of the preprocessor. For example, an electron requirement needs the calorimeter EM preprocessor, the L2CTT, and possibly the L2PS preprocessor.

In the L2CTT, receipt of a *UBS* qualifier will result in readout to L3 of the L2CTT inputs, and possible additional output information beyond that sent to L2G. The standard output to L2G is not changed by the *UBS* bit.

The *Forced Write* qualifier is intended to provide a mechanism for special runs meant to study new L2 triggers. It has exactly the same effect as the *UBS* bit in L2, however, the behavior in L3 is different.

After receipt of an event with a *Collect Status* qualifier, the L2CTT (and all other L2 crates) will capture their scalers and other monitoring information and place it where TCC can retrieve it for serving with monitoring (to be read out via the Bit3). The scalers will be captured after processing of the event is completed, so that scaler information should be exactly matched between the L1 and L2 HWFW, the preprocessors and L2G. The monitoring blocks will be tagged with the L1 crossing number of the event with the *Collect Status* qualifier, so TCC can assemble a consistent set of statistics. *Collect Status* qualifiers will be generated once every 5 seconds (0.2 Hz).

Command	Description
SCL_READY	Serial Command Link Ready Status
SCL_SYNCERROR	Serial Command Link Synchronization Error
SCL_DATAERROR	Serial Command Link Data Error
SCL_ACK	Acknowledge and Clear SCL Error Flags
CLK_53	53 MHz Clock
CLK_7	7.5 MHz Clock
CURRENT_TURN[15..0]	Current Turn Number
CURRENT_BX[7..0]	Current BX Number in this Turn
FIRST_PERIOD	First Period in a Turn Marker
BEAM_PERIOD	Period with Beam Marker
SYNC_GAP	Sync Gap Marker (No L1 Accepts)
COSMIC_GAP	Cosmic Gap Marker (Only Cosmic L1 Accepts)
SPARE_PERIOD	Spare Period Marker
L1_PERIOD	Period with L1 Accept Issued
L1_ACCEPT	L1 Accept in this Geo Section
L1_TURN[15..0]	Level 1 Turn Number
L1_BX[7..0]	Level 1 BX Number in this Turn
L1_Qual[15..0]	L1 Accept Qualifiers/Geo Section L3 Transfer Number
L2_PERIOD	Period with L2 Decision Issue
L2_REJECT	This Geo Section L2 Reject
L2_ACCEPT	This Geo Section L2 Accept
INIT_SECTION	Initialize Geographic Section Flag
L1_NEEDED	Tells a L2 preprocessor if event needs anaylsis.
L1_BUSY	Busy L1 Status
L2_BUSY	Busy L2 Status
L1_ERROR	Error L1 Flag
L2_ERROR	Error L2 Flag
INIT_ACK	Init_Ack Signal to Hub-End
SYNC_LOST	SCL Receiver Synchronization Lost
SPARE_STATUS[1..0]	Spare Status Signals to Hub-End

Table 12: The commands from the Serial Command Link.

B Appendix: L2G Data Transfer Requirements

We itemize here the requirements for the data transfer between L2CTT and L2G. We point out the specific requirements where appropriate.

- The smallest fragment size for a variable within a track object is 1 byte.
- The track objects's total data size must be an integer number of longwords (1 longword = 4 bytes = 32 bits); a L2CTT track object is 2 longwords.
- Multi-byte fields do not cross boundaries between longword boundaries.
- The data must prefaced with a header which is an integer number of longwords in length; the L2CTT header is 3 longwords.
- The data transmission will be completed with a trailer which is an integer number of longwords; the L2CTT trailer is 1 longword.
- Some number of bytes will be added after the trailer to make the total number of longwords divisible by 4 since the Magic Bus is a 16B wide bus. Repeated footers are sent to fulfill this criteria.
- Where possible each track object will be identified by rapidity, azimuth and E_T in that order[7]
- Tracks objects will be sorted in descending P_T order[8].
- The least significant bit for rapidity will be 0.05, for azimuth $2\pi/160$, and for transverse momentum 0.25 GeV/c.
- Azimuth will be reported in standard DØ coordinates. Since the charged particles curve in the magnetic field, the reported values of ϕ will be extrapolated to two different interesting radii. These are at $r = 91.6$ cm and $r = 0$. The first radius is the center of EM3 in the central calorimeter, or showermax, (used in electron and muon matching) and will be referred to as ϕ_{EM3}. The extrapolation back to $r = 0$ is useful for for physics object processing (such as J/ψ mass reconstruction) and is referred to as ϕ_0.

[7]Note that this only partially applies to the L2CTT until the STT is used as an input.

[8]Note that when the input is from the STT, there will be two separate lists from the two worker Alphas: one by P_T and one by impact parameter significance.

C L1 to L2 $\mathrm{P_T}$ Conversion with the Offset Method

In this appendix we give details on the Offset to $\mathrm{P_T}$ conversion. To convert from offset to $\mathrm{P_T}$ we use the 4 bits of offset information as well as the H layer fiber number. This conversion is subtle. For example, if at the A layer the found track passes through fiber 0 and at the H layer the fiber hit is fiber 4 then the offset will be reported as 1 and the sign will be equal to 1. How did we determine this? H layer fibers of 0, 1 and 2 have an offset of 0 from A layer fiber 0. Similarly H layer fibers 3, 4 and 5 have an offset of 0 from A layer fiber 1 (they have an offset of 1 from A layer fiber 0). To complete the loop, H layer fibers 6 and 7 have an offset of 0 from A=2 and H layer fiber numbers 8, 9 and 10 are associated with an A=3. This repeats every 4 A layer fiber numbers. Thus, given an H layer fiber number and an offset one can determine the ϕ_A.

To convert the offset (and thus $\Delta\phi$) to a $\mathrm{P_T}$ we use the equations:

$$\sin(\phi_{\mathrm{A}} - \phi_0) = \frac{0.15 \cdot \mathrm{B} \cdot \mathrm{R_A}}{P_T} \qquad \sin(\phi_{\mathrm{H}} - \phi_0) = \frac{0.15 \cdot \mathrm{B} \cdot \mathrm{R_H}}{P_T} \tag{3}$$

where B is the magnetic field in Tesla (= 2 Tesla), R is the distance from the center of the detector (R_A = 0.2008 m and R_H = 0.5152 m) and ϕ_A and ϕ_H are the values of ϕ at the "A" layer and "H" layer of the fiber tracker respectively. We have two equations and two unknowns ($\mathrm{P_T}$ and ϕ) so we can solve for both. This is done by expanding the sine functions to first order. This is acceptable because $\phi_H - \phi_0$ and $\phi_A - \phi_0$ are small compared to the desired precision here. After solving we obtain the equations (angles are in radians):

$$\phi_0 = \frac{\phi_A - 0.38 \times \phi_H}{0.62} \qquad \mathrm{P_T} = \frac{0.094}{\phi_H - \phi_A} \tag{4}$$

Now we give an example of offset conversion. Using our previous example with offset = 1 and H = 4 and sign = 1, we can deduce that the fibers hit were A=0 and H=4. Since there are 1,280 and 3520 fibers in the A and H layers respectively, the fiber widths are 4.91 mrad and 1.78 mrad respectively. Thus, the center of the A=0 fiber is at $\phi(A=0) = 2.45$ mrad (relative to the sector boundary) and the center of the H=4 fiber is at $\phi(H=4) = 8.03$ mrad for a $\Delta\phi$= 3.79 mrad. Plugging this into Equation 4 we find:

$$\begin{aligned}
\Delta\phi &= \phi(H=4) - \phi(A=0) \\
&= 0.00803 - 0.00245 \\
&= 0.00558 \\
\mathrm{P_T} &= \frac{0.15 \cdot 2 \cdot 0.32}{0.00558} = 17.21 \text{ GeV}
\end{aligned}$$

Note that the value sent to L2G would be 17.25 GeV since the least significant bit corresponds to 0.25 GeV. Before any events are run, the L2CTT initializes itself by calculating the $\mathrm{P_T}$ for all offset and H combinations and putting them into a lookup table.

Using the Worker prototype code we have run a toy simulation of the L1CFT (see Section 8)in which the parameters of a particle trajectory are converted into FE number, H fiber number, offset, and sign by simply using the A and H layer fiber numbers the track passes through. Figure 11 shows the reported offset as a function of input $\mathrm{P_T}$.

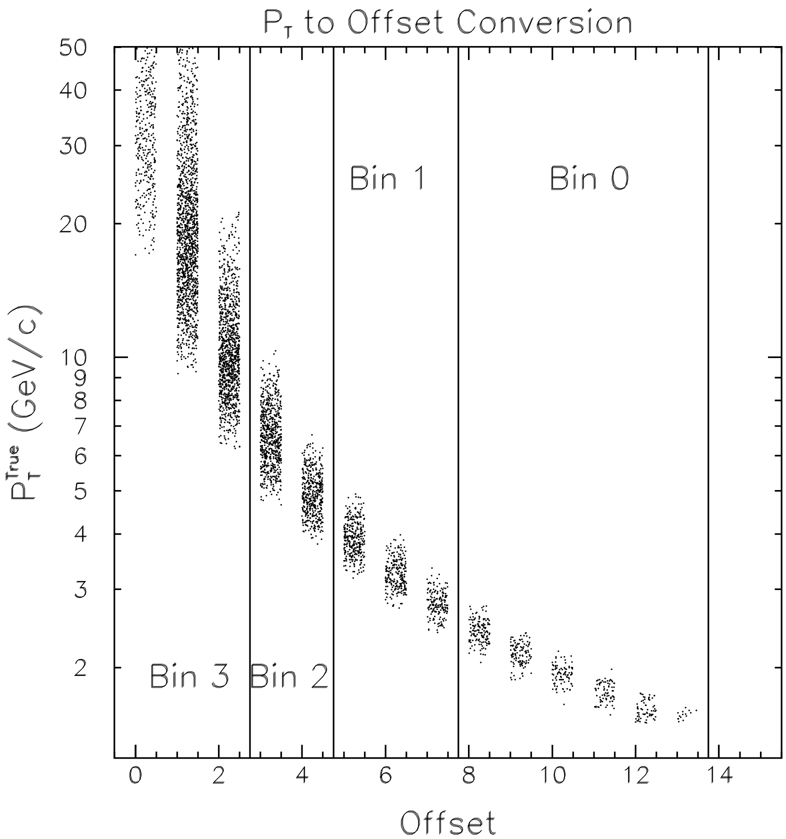

Figure 11: P_T vs. Offset for the toy simulation of L1CFT. Note that the P_T is not a function of offset. Said differently, not all tracks with an offset of 2 are of higher P_T than have an offset of 3.

The actual values of the offsets are correlated with extended P_T reported by L1. As can be seen from Fig. 11 a given P_T value may correspond to different offset values. Therefore, in order to optimize the momentum resolution, for each extended P_T an optimum offset must be evaluated. We used Monte Carlo single muon events with P_T in the range 1.5 - 10 GeV to obtain optimized offsets for each P_T bin and extended P_T. For each track an ideal offset was evaluated. With the ideal offset the trigger momentum equals the input P_T. For each pair of P_T bin and extended P_T the average of ideal offsets was evaluated. These average values were assigned to optimized offsets, and are shown in Table 13.

The offset method is useful only when P_T is low enough (< 5 GeV). In terms of offsets, this method gives better momentum resolution if offsets are greater than 4. Figure 12 shows the P_T resolution at L2CTT for a single muon sample when trigger track momenta were calculated with optimized offsets.

Bin #	Extended P_T	Offset
2	0	4.00
2	1	4.58
2	2	5.58
2	3	6.57
3	0	7.58
3	1	8.61
3	2	9.60
3	3	10.66
3	4	11.55
3	5	12.22
3	6	12.80
3	7	13.50

Table 13: Values of optimized offsets corresponding to P_T bin and extended P_T pairs.

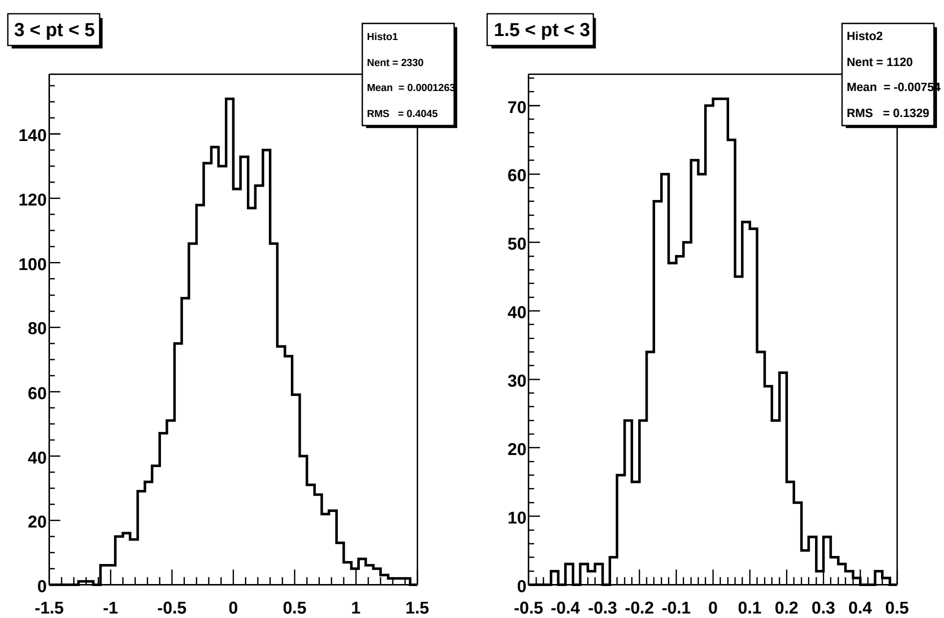

Figure 12: Plots of $(P_T)_{MC} - (P_T)_{trigger}$ for single muon events (bins 2 and 3). Trigger track momenta were calculated using optimized offsets.